Final Project

Christophe Hunt

May 13, 2017

Contents

1 Variable

Pick one of the quantitative independent variables from the training data set (train.csv), and define that variable as X.

Pick SalePrice as the dependent variable, and define it as Y for the next analysis.

1.1 Variable Picked

> The variable we will set to X is LotArea, which is defined as the Lot size in square feet. I chose LotArea because an anecdotal assumption is that the larger the lot size is the higher the sale price. However, living in NYC, I know that tiny lots in very desirable places have sold for a high price so I believe there may be some interesting varability.

```r
library(tidyverse)
train.df <- as_tibble(read.csv(paste("https://raw.githubusercontent.com/", "ChristopheHunt/",
    "MSDA---Coursework/master", "/Data%20605/Final%20Project/train.csv", sep = "")))
```

```r
sub.train.df <- train.df[, c("SalePrice", "LotArea")]
```

2 Probability

Calculate as a minimum the below probabilities a through c.

Assume the small letter “x” is estimated as the 4th quartile of the X variable, and the small letter “y” is estimated as the 2nd quartile of the Y variable. Interpret the meaning of all probabilities.

```
prob.x <- list(qrt = as.numeric(quantile(sub.train.df$LotArea)[4]))

prob.y <- list(qrt = as.numeric(quantile(sub.train.df$SalePrice)[2]))
```

```
prob.y.x <- sub.train.df %>% mutate(greaterLotArea = ifelse(LotArea >= prob.x$qrt, 1, 0),
                                     lesserLotArea = ifelse(LotArea < prob.x$qrt, 1, 0),
                                     greaterSalePrice = ifelse(SalePrice >= prob.y$qrt,1, 0),
                                     lesserSalePrice = ifelse(SalePrice < prob.y$qrt,1, 0))
```

2.1 a. $P(X > x|Y > y)$

```
a <- (sum(ifelse(prob.y.x$greaterLotArea == 1 &
                   prob.y.x$greaterSalePrice == 1, 1, 0))
      / nrow(prob.y.x)) / ((sum(prob.y.x$greaterLotArea) / nrow(prob.y.x)))
a
```

```
## [1] 0.9369863
```

2.2 b. $P(X > x, Y > y)$

```
b <- sum(ifelse(prob.y.x$greaterLotArea == 1 &
                  prob.y.x$greaterSalePrice == 1, 1, 0))/nrow(prob.y.x)
b
```

```
## [1] 0.2342466
```

2.3 c. $P(X < x|Y > y)$

```
c <- (sum(ifelse(prob.y.x$lesserLotArea == 1 &
                   prob.y.x$greaterSalePrice == 1, 1, 0))
      / nrow(prob.y.x)) / ((sum(prob.y.x$lesserLotArea) / nrow(prob.y.x)))
c
```

```
## [1] 0.6876712
```

Does splitting the training data in this fashion make them independent?

In other words, does $P(X|Y) = P(X)P(Y)$?

> I am understanding this to mean does the probability of X>x given Y>y, which was answered for in part a. above, equal the probability of X>x mutiplied by Y>y

2.4 Mathematical Check for $P(X|Y) = P(X)P(Y)$

```
X <- sum(prob.y.x$greaterLotArea)/ nrow(prob.y.x)
Y <- sum(prob.y.x$greaterSalePrice) / nrow(prob.y.x)
X * Y
```

```
## [1] 0.1875
```

```
a == (X * Y)
```

```
## [1] FALSE
```

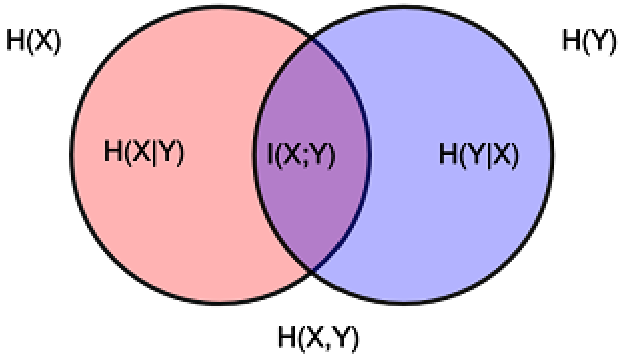
2.5 Chi Square test for association.

```
prob.table <- as.data.frame(rbind(cbind(sum(prob.y.x$lesserLotArea), sum(prob.y.x$greaterLotArea)), cbi
chisq.test(prob.table)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  prob.table
## X-squared = 728, df = 1, p-value < 2.2e-16
```

> We see that the p-value is quite low, lower than the assumptive .05, so we therefore reject the null hypothesis that the values are independent of each other.

The below venn diagram from Wikipedia may provide a clearer understanding of the differences in these measures:



[ˆ4]

[ˆ4] By KonradVoelkel (Own work) [Public domain], via Wikimedia Commons

3 Descriptive and Inferential Statistics.

Provide univariate descriptive statistics and appropriate plots for both variables.

```
description <- describe(sub.train.df["LotArea"])
latex(description, file = '')
```

sub.train.df["LotArea"]
1 Variables 1460 Observations

```
LotArea
```

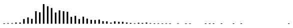
```
     n   missing  distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75      .90      .95
  1460         0      1073        1    10517     5718     3312     5000     7554     9478    11602    14382    17401

lowest :    1300    1477    1491    1526    1533, highest:  70761 115149 159000 164660 215245
```

The histogram shows a right skewed distribution, which is not unexpected since houses in cities would likely have similar lot areas versus the rarer instances of large variable lot areas.

```r
description <- describe(sub.train.df["SalePrice"])
latex(description, file = '')
```

sub.train.df["SalePrice"]
1 Variables   1460 Observations

```
SalePrice
```

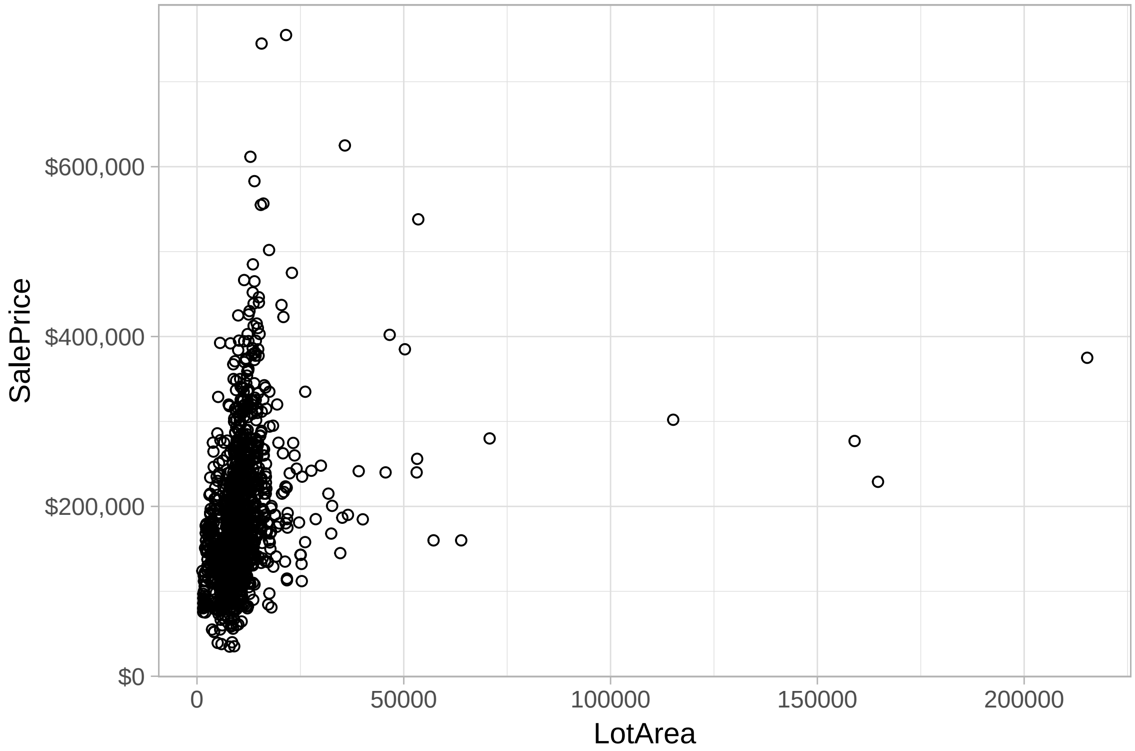


```
     n   missing  distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75      .90      .95
  1460         0       663        1   180921    81086    88000   106475   129975   163000   214000   278000   326100

lowest :   34900   35311   37900   39300   40000, highest: 582933 611657 625000 745000 755000
```

As we can see from the histogram the shape of the data is near normal. It is interesting to think about how lot area does not follow the same shape, this would hold with our original assumption that where the house is located has more impact than the size of the lot area.

Provide a scatterplot of X and Y.

```r
ggplot(sub.train.df, aes(x = LotArea, y = SalePrice)) + geom_point(shape = 1) +
    theme_light() + scale_y_continuous(labels = dollar)
```



Transform both variables simultaneously using Box-Cox transformations.

I am using the `BoxCox.lambda` function from the `forecast` package to determine the necessary transformations for the two variables.

```
library(forecast)
library(knitr)
l1 <- BoxCox.lambda(as.numeric(sub.train.df$SalePrice))
l2 <- BoxCox.lambda(as.numeric(sub.train.df$LotArea))

lamdas <- c(l1, l2)
Variables <- c("SalePrice", "LotArea")
dfBoxCox <- as.data.frame(cbind(round(as.numeric(lamdas),4), Variables))
colnames(dfBoxCox) <- c("$\\lambda$", "Variables")
kable(dfBoxCox, align = c("c", "c"))
```

λ	Variables
-0.3308	SalePrice
-0.1268	LotArea

Common Box-Cox Transformations[1] [2]

λ	Y'
-0.5	$Y^{-0.5} = \frac{1}{\sqrt{(Y)}}$
0	$\log(Y)$
.25	$\sqrt[4]{Y}$

Lambda values were truncated to the nearest tenth that match a common transformation as per the below table.

variable	variable transformation
SalePrice	$SalePrice^{-0.5}$
LotArea	$log(LotArea)$

3.1 Correlation Analysis

Using the transformed variables, run a correlation analysis and interpret.

```
sub.train.df.trans <- sub.train.df %>%
                      mutate(SalePrice = SalePrice^(-.5),
                             LotArea = log(LotArea))

sub.train.cor <- cor.test(sub.train.df.trans$SalePrice,
                          sub.train.df.trans$LotArea,
                          method = "pearson", conf.level = .99)
sub.train.cor
```

```
##
##  Pearson's product-moment correlation
##
## data:  sub.train.df.trans$SalePrice and sub.train.df.trans$LotArea
## t = -15.968, df = 1458, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
```

[1] Osborne, Jason W. “Improving your data transformations: Applying the Box-Cox transformation.” Practical Assessment, Research & Evaluation 15.12 (2010): 1-9.

[2] By Understanding Both the Concept of Transformation and the Box-Cox Method, Practitioners Will Be Better Prepared to Work with Non-normal Data. . “Making Data Normal Using Box-Cox Power Transformation.” ISixSigma. N.p., n.d. Web. 29 Oct. 2016.

```
## 99 percent confidence interval:
##  -0.4417063 -0.3269282
## sample estimates:
##        cor
## -0.3858091
```

> The p-value of the correlation test is 2.2e-16 which is less than the significance level of alpha at .05. We are using the standard alpha as there is no indication another any other value for alpha should be used. We can therefore say that the log of lot size and sale price raised to the -.5 power are significantly correlated with a negative correlation coefficient of -0.386.

Test the hypothesis that the correlation between these variables is 0 and provide a 99% confidence interval.

> The correlation test has specifically done that for us and we can safely reject the null hypothesis as we see that our 99% confidence interval exists at the values (-0.441, -0.327) with a p-value < 2.2e-16.

Discuss the meaning of your analysis.

> This means two possible things could have occured, there is no correlation and this data set is pulled from an unusual set of house sales. Or, more likely with the values obtained, our assumption of 0 correlation is incorect and we have obtained a very typical data set and must reject the null hypothesis because correlation does exist.

4 Linear Algebra and Correlation.

```
A <- cor(sub.train.df.trans)
kable(A)
```

	SalePrice	LotArea
SalePrice	1.0000000	-0.3858091
LotArea	-0.3858091	1.0000000

Invert your correlation matrix.(This is known as the precision matrix and contains variance inflation factors on the diagonal.)

```
B <- solve(A)
kable(B)
```

	SalePrice	LotArea
SalePrice	1.1748792	0.4532792
LotArea	0.4532792	1.1748792

Multiply the correlation matrix by the precision matrix, and then multiply the precision matrix by the correlation matrix.

```
corr.by.pre.M <- A %*% B
kable(corr.by.pre.M)
```

	SalePrice	LotArea
SalePrice	1	0

	SalePrice	LotArea
LotArea	0	1

```
pre.by.corr.M <- B %*% A
kable(pre.by.corr.M)
```

	SalePrice	LotArea
SalePrice	1	0
LotArea	0	1

5 Calculus-Based Probability & Statistics

Many times, it makes sense to fit a closed form distribution to data. For your non-transformed independent variable, location shift it so that the minimum value is above zero.

```
min(sub.train.df$LotArea)
```

```
[1] 1300
```

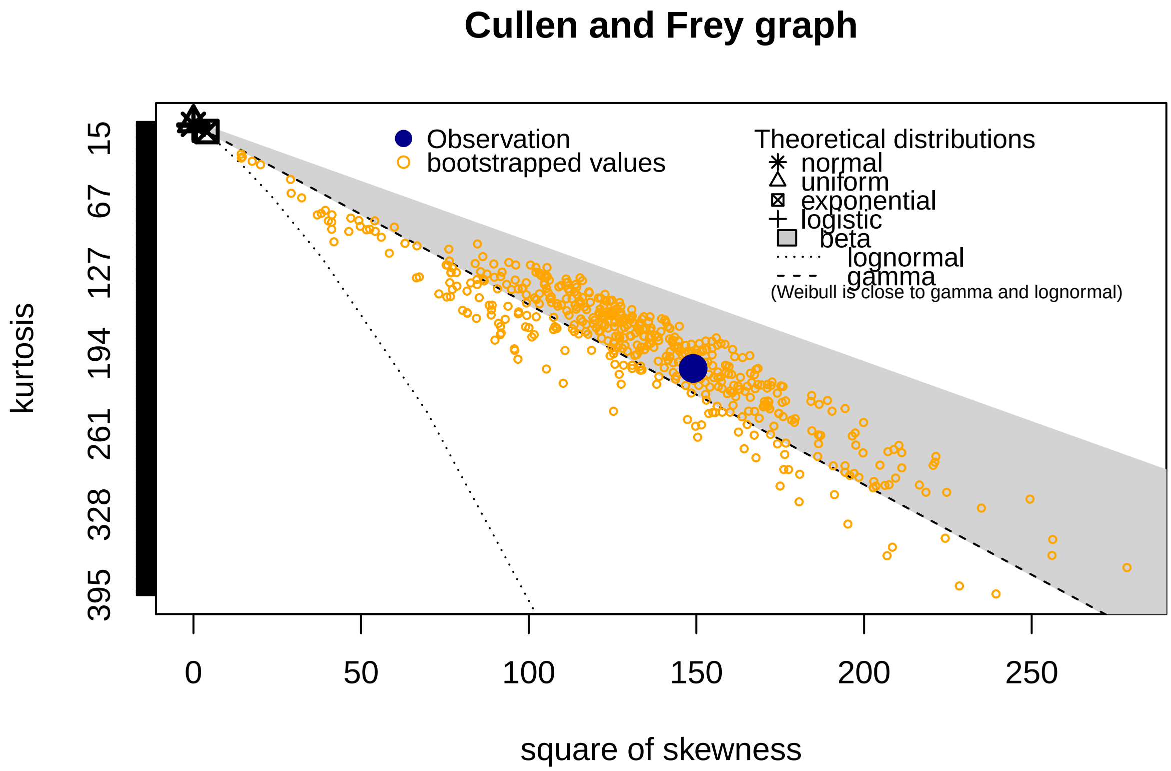
> For the independent variable chosen, there are no zero values observed. This makes sense as we would expect the lot area to have some value and I would expect it to never be unobserved (an assumption that at least estimates would be used without a true figure).
>
> However, if a shift was required something like the below could be used.

```
shift <- sub.train.df$LotArea + 1
```

Then load the MASS package and run fitdistr to fit a density function of your choice. (See https://stat.ethz.ch/R-manual/R-devel/library/MASS/html/fitdistr.html).

> First lets look at what distrubtion would best fit our data.

```
library(fitdistrplus)
descdist(sub.train.df$LotArea, discrete=FALSE, boot=500)
```



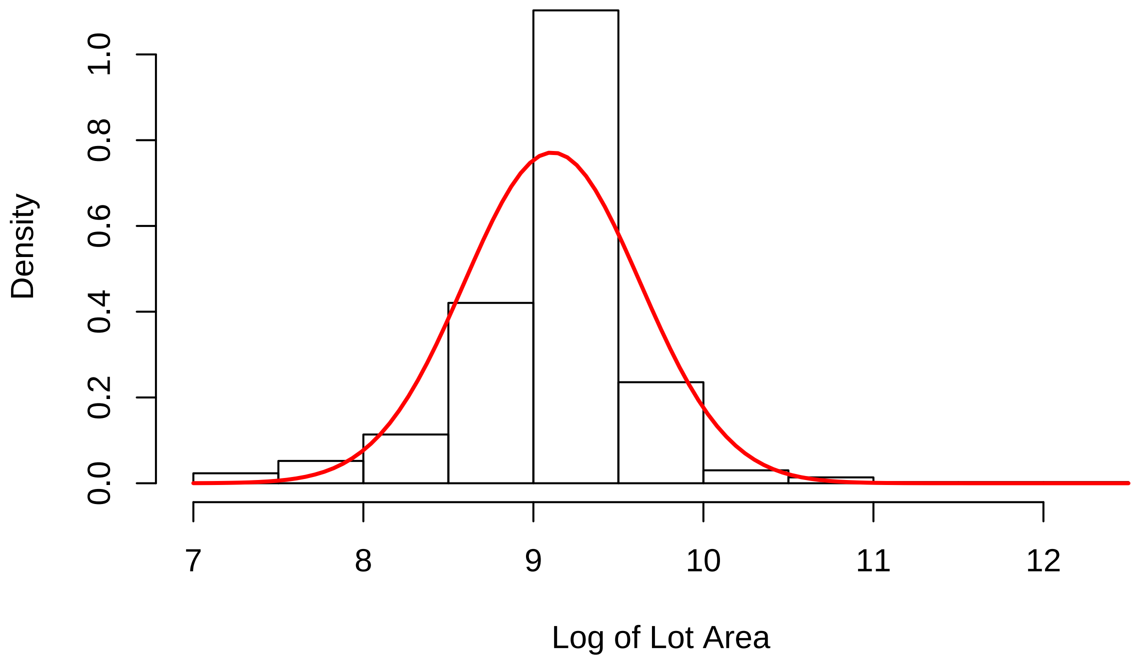
```
## summary statistics
## ------
## min:  1300   max:  215245
## median:  9478.5
## mean:  10516.83
## estimated sd:  9981.265
## estimated skewness:  12.20769
## estimated kurtosis:  206.2433
```

There were too many issues in attempting to fit the beta distribution so the next best theoretical distribution was used - log normal.

```
library(MASS)
fit.log <- fitdistr(sub.train.df$LotArea, densfun = "log-normal")
fit.log
```

```
##      meanlog        sdlog
##   9.110838240   0.517270830
##  (0.013537596) (0.009572526)
```

```
hist(log(sub.train.df$LotArea), prob=TRUE, xlab = "Log of Lot Area", main = "")
curve(dnorm(x, fit.log$estimate[1], fit.log$estimate[2]), col="red", lwd=2, add=T)
```

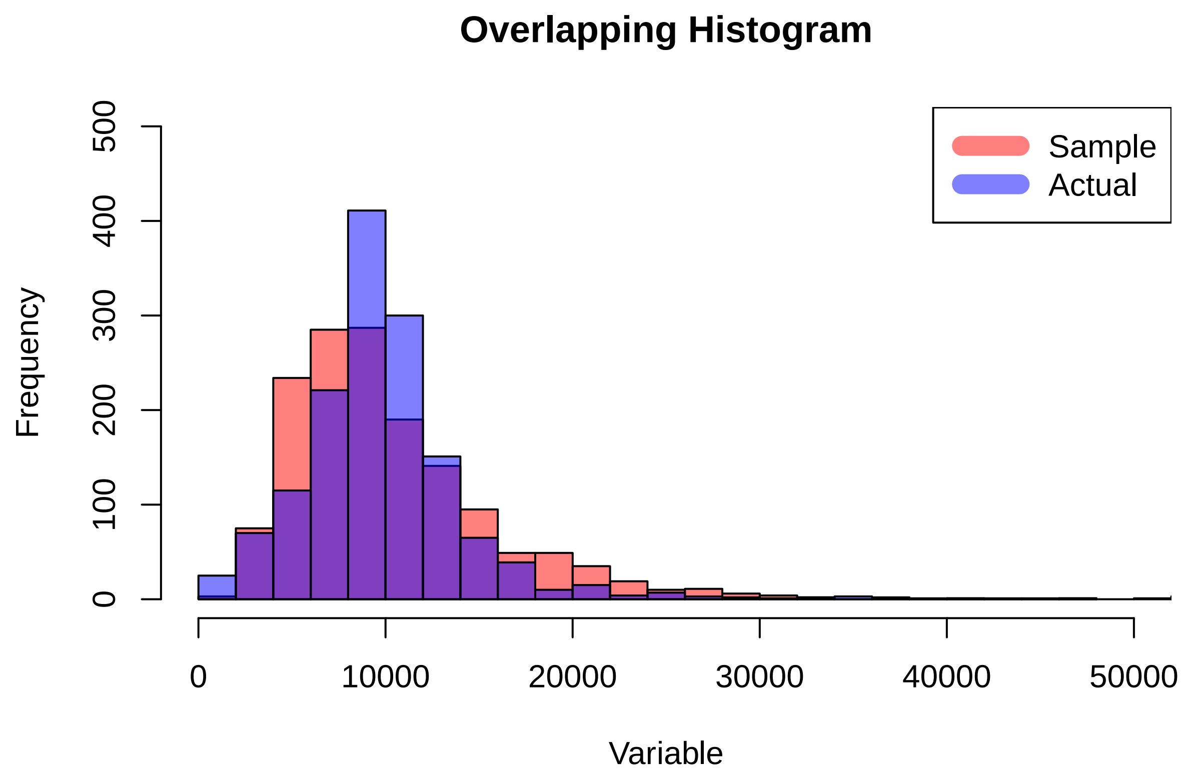


From our density plot, the distribution looks quite good.

Find the optimal value of the parameters for this distribution, and then take 1000 samples from this distribution (e.g., rexp(1000) for an exponential).

```r
set.seed(1234)
sample <- rlnorm(1000, meanlog = fit.log$estimate[1], sdlog = fit.log$estimate[2])
```

```r
hist(sample, pch = 20, breaks = 25, col = rgb(1,0,0,0.5), xlim = c(0,50000), ylim = c(0,500), main = 'O
hist(sub.train.df$LotArea, pch = 20, breaks = 100, col = rgb(0,0,1,0.5), add = T)
#https://www.r-bloggers.com/overlapping-histogram-in-r/
legend("topright", c("Sample", "Actual"), col=c(rgb(1,0,0,0.5), rgb(0,0,1,0.5)), lwd=10)
```



Plot a histogram and compare it with a histogram of your non-transformed original variable.

> It is clear that the distributions are very similar. Plotting them overlapping gives a clear visual of how similar the distributions, note that x has been limited and does not extend out for extreme values of x.

6 Modeling

Build some type of regression model and submit your model to the competition board.

```
library(caret)
#set up dummy columns
dummies <- dummyVars(SalePrice ~ ., data = train.df)
train.df.dum <- as.data.frame(predict(dummies, newdata = train.df))
train.df.dum <- as.data.frame(train.df.dum %>% dplyr::select(-starts_with(c("Alley"))))
```

6.1 EVAL SET TO FALSE

```
library(missForest)
registerDoParallel(cl = makeCluster(25), cores = 100)
set.seed(1234)
train.df.dum.imp <- train.df.dum %>% missForest(maxiter = 10, ntree = 100, replace = TRUE, parallelize
write.csv(train.df.dum.imp$ximp,"imputed_training_data.csv", row.names = FALSE) #wrote imputed_data to
```

```r
library(caret)
sbf(train.df$LotArea, train.df$SalePrice)

fit <- sbf(form = SalePrice ~ .,
           data = train.df,
           method = "svmLinear",
           trControl = trainControl(method = "none",
                                    classProbs = TRUE),
           preProc = c("center", "scale"))
```

```r
library(leaps)
library(MASS)
regsubsets(SalePrice ~ Id + MSSubClass + MSZoning  + LotArea + Street + LotShape + LandContour + Utilit
```

```r
library(lme4)
library(nlme)
library(arm)
#testing the random effect
#a first model
mod1<-lme(SalePrice~LotArea+SaleCondition,data=train.df, random=~1|SaleCondition, method="REML")

anova(mod1)
predict(mod1, train.df)
```

Provide your complete model summary and results with analysis.

Report your Kaggle.com user name and score.